Implementasi Skema Shamir's Secret Sharing untuk Desentralisasi Penyimpanan *Seed Phrase* Dompet Kripto

Diero Arga Purnama - 13522056
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13522056@std.stei.itb.ac.id, diero.purnama@gmail.com

***Abstrak*—Penyimpanan *seed* phrase dompet kripto secara konvensional pada suatu media fisik atau digital menciptakan risiko *Single Point of Failure*, di mana kerusakan atau kehilangan media tersebut dapat menyebabkan kehilangan aset permanen. Penelitian ini mengimplementasikan skema Shamir's Secret Sharing (SSS) untuk mendesentralisasi penyimpanan kunci dengan memecahnya menjadi beberapa bagian (*shares*). Menggunakan aritmatika *Finite Field* dengan modulus Bilangan Prima Mersenne ke-16 ($2^{2203}-1$), sistem dirancang untuk mengakomodasi panjang bit *seed phrase* standar tanpa risiko *overflow*. Hasil pengujian menunjukkan bahwa rekonstruksi frasa hanya berhasil jika ambang batas *shares* terpenuhi, menjamin keamanan dan ketersediaan akses aset digital yang tangguh terhadap kehilangan media penyimpanan.**

***Kata Kunci*—Shamir's Secret Sharing, Kriptografi, *Seed Phrase*, Desentralisasi Kunci.**

I. PENDAHULUAN

Dalam era transformasi digital, paradigma penyimpanan aset telah bergeser dari sistem perbankan terpusat menuju sistem terdesentralisasi atau *Decentralized Finance* (DeFi). Berbeda dengan sistem perbankan yang bertindak sebagai pihak ketiga terpercaya, sistem ini menuntut pengguna untuk memegang kendali penuh atas akses dan keamanan aset mereka. Akses terhadap dompet digital ini umumnya bergantung pada kunci privat yang direpresentasikan dalam bentuk *seed phrase* yang terdiri dari 12 hingga 24 kata. Namun, mekanisme ini memiliki kelemahan besar jika dikelola secara konvensional. Penyimpanan *seed phrase* sepenuhnya pada satu media fisik (seperti kertas) atau digital menciptakan risiko *Single Point of Failure* (SPoF). Apabila media penyimpanan hilang, dicuri, ataupun rusak akibat bencana, maka seluruh aset yang tersimpan akan hilang secara permanen tanpa adanya mekanisme pemulihan seperti fitur "*Forgot Password*" pada bank.

Oleh karena itu, dibutuhkan strategi penyimpanan yang dapat memitigasi risiko tersebut tanpa mengurangi integritas data. Shamir's Secret Sharing (SSS) merupakan algoritma kriptografi yang menawarkan solusi ideal untuk permasalahan ini melalui skema pembagian data rahasia. Algoritma ini memungkinkan *seed phrase* dipecah menjadi beberapa bagian (*shares*) yang didistribusikan ke lokasi berbeda, di mana rekonstruksi frasa hanya dapat dilakukan jika jumlah bagian yang terkumpul mencapai ambang batas yang telah ditentukan. Dengan memanfaatkan skema ini, kehilangan satu bagian kunci tidak akan menyebabkan hilangnya akses secara permanen, sekaligus mencegah pihak yang tidak berwenang mengakses dompet hanya dengan menemukan satu bagian kunci.

Makalah ini bertujuan untuk mengimplementasikan skema SSS menggunakan aritmatika *Finite Field* berbasis Bilangan Prima Mersenne ke-16 ($2^{2203}-1$) untuk menangani *seed phrase* panjang, serta menganalisis efektivitasnya sebagai metode penyimpanan kunci mandiri yang aman dan terdesentralisasi. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi praktis dalam penyediaan mekanisme penyimpanan aset digital yang tangguh terhadap risiko kegagalan fisik maupun pencurian data.

II. LANDASAN TEORI

A. Seed Phrase

Seed Phrase merupakan sebuah rangkaian kata acak yang terdiri dari 12, 18, atau 24 kata yang akan dihasilkan ketika membuat dompet kripto baru [1]. Frasa ini berfungsi untuk menyimpan data yang diperlukan untuk mengakses dompet tersebut [2].

Oleh karena itu, *seed phrase* dapat diumpamakan sebagai kunci bagi seluruh aset pengguna. Siapa pun yang memiliki frasa ini memegang kendali penuh dan dapat memindahkan dana dari dompet tersebut. *Seed phrase* juga berfungsi untuk melakukan pemulihan akses jika perangkat pengguna rusak ataupun hilang. Berbeda dengan kata sandi pada layanan perbankan konvensional yang dapat diatur ulang melalui bantuan pihak ketiga, kehilangan *seed phrase* berarti kehilangan aset secara permanen [1].

Seed Phrase merupakan sebuah rangkaian kata acak

yang terdiri dari 12, 18, atau 24 kata yang akan dihasilkan ketika membuat dompet kripto baru [1]. Frasa ini berfungsi untuk menyimpan data yang diperlukan untuk mengakses dompet tersebut [2].

Pemilihan kata dalam *seed phrase* mengacu pada protokol standar industri yang dikenal sebagai BIP-39 (*Bitcoin Improvement Proposal* 39). Protokol ini menetapkan daftar kata yang terdiri dari 2048 kata bahasa inggris yang spesifik [3]. Daftar kata ini berisi kata-kata yang telah dipilih secara khusus untuk menghindari kemiripan ejaan dan pelafalan untuk meminimalkan kesalahan pencatatan oleh pengguna. Dalam implementasi sistem ini, standar BIP-39 dijadikan sebagai acuan untuk menganalisis struktur kata dan menghitung estimasi kebutuhan ruang penyimpanan. Analisis dilakukan terhadap daftar kata untuk menentukan panjang karakter maksimum yang mungkin terbentuk. Perhitungan ini krusial untuk menentukan batas atas ukuran data, memastikan bahwa bilangan prima yang dipilih mampu menampung konversi teks ke integer tanpa risiko *overflow*.

B. Shamir's Secret Sharing

Shamir's Secret Sharing (SSS) adalah metode kriptografi yang dikenalkan oleh Adi Shamir pada tahun 1979 untuk memecahkan masalah pengamanan data sensitif melalui distribusi kepercayaan. SSS memanfaatkan skema ambang $(t,\ n)$, di mana sebuah rahasia S dibagi menjadi n bagian atau *shares*. Dalam skema ini, rahasia S hanya dapat direkonstruksi kembali jika setidaknya terdapat t bagian yang digabungkan. Jika jumlah bagian yang tersedia kurang dari t, maka rahasia S tidak dapat direkonstruksi sama sekali, yang merupakan keunggulan utama dari skema ini [4].

Secara matematis, algoritma ini dibangun di atas konsep interpolasi polinomial. Algoritma harus terlebih dahulu memilih sebuah bilangan prima p yang nilainya lebih besar dari rahasia S dan jumlah partisipan n ($p > S$ dan $p > n$).

Proses pembentukan *share* dilakukan dengan membangkitkan sebuah polinomial acak $f(x)$ berderajat $t-1$ sebagai berikut:

$$f(x) \equiv S + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \pmod{p} \tag{1}$$

Di mana S adalah rahasia yang ditempatkan pada konstanta a_0 (atau $f(0)$), dan a_1, …, a_{t-1} adalah koefisien acak yang dipilih dari elemen modulus p. Setiap partisipan ke-i kemudian akan menerima *shares* dalam bentuk pasangan titik (x_i, y_i) yang dihasilkan dengan mengevaluasi fungsi tersebut pada nilai x yang berbeda.

$$y_i \equiv f(x_i) \pmod{p} \tag{2}$$

Meskipun setiap partisipan memegang satu titik dari kurva, bentuk utuh polinomial tidak dapat diketahui kecuali t buah titik dikumpulkan untuk melakukan interpolasi.

C. Interpolasi Lagrange

Interpolasi Lagrange adalah metode untuk mencari nilai dari suatu fungsi pada suatu titik untuk suatu fungsi yang tidak diketahui [7]. Jika diberikan sekumpulan titik koordinat (x_1, y_1), (x_2, y_2), …, (x_t, y_t), rumus umum Polinomial Lagrange didefinisikan sebagai kombinasi linier dari $L_j(x)$.

$$P(x) = \sum_{j=1}^{t} y_j L_j(x) \tag{3}$$

Di mana $L_j(x)$ merupakan basis polinomial yang dibentuk dari rumus berikut:

$$L_j(x) = \prod_{i=1,\ i \neq j}^{t} \frac{x - x_i}{x_j - x_i} \tag{4}$$

Basis $L_j(x)$ dirancang sedemikian rupa sehingga bernilai 1 pada titik x_j dan bernilai 0 pada titik-titik x_i lainnya. Dalam konteks pemulihan data pada skema SSS, tujuan utamanya bukan untuk mencari persamaan polinomial secara utuh, melainkan hanya menemukan nilai konstanta $f(0)$. Oleh karena itu, persamaan tersebut dapat disederhanakan dengan mensubstitusikan nilai $x = 0$. Hasil evaluasi $P(0)$ ini akan sama dengan nilai $f(0)$ pada polinomial pembentuk (persamaan 1), yang merupakan nilai rahasia S.

D. Aritmatika Finite Field

Untuk menjamin presisi komputasi kriptografi, sistem ini tidak beroperasi pada himpunan bilangan riil, melainkan pada struktur aljabar abstrak yang disebut *Galois Field*. Secara spesifik, penelitian ini menggunakan kelas medan prima F_p, yang didefinisikan sebagai himpunan bilangan bulat {0, 1, 2, ..., $p-1$} di mana p adalah bilangan prima [8]. Dalam struktur ini, seluruh operasi penjumlahan dan perkalian dilakukan dalam modulus p, yang menjamin hasil perhitungan selalu berada dalam lingkup himpunan tersebut.

E. Bilangan Prima Mersenne

Dalam matematika, bilangan Mersenne merupakan sebuah bilangan yang dapat dinyatakan dalam bentuk $Mn = 2^n - 1$, di mana n adalah bilangan bulat positif [5]. Bilangan prima Mersenne adalah bilangan Mersenne yang terbukti merupakan bilangan prima [6].

Bilangan prima Mersenne memiliki peran yang signifikan dalam dunia komputasi karena struktur binernya. Karena bentuknya yang merupakan pangkat dua dikurangi satu, representasi biner dari bilangan ini terdiri sepenuhnya dari digit 1. Karakteristik ini memungkinkan operasi aritmatika modular pada komputer biner dilakukan

dengan efisiensi tinggi dibandingkan dengan bilangan prima sembarang lainnya. Dalam penelitian ini, aritmatika *Finite Field* dibangun menggunakan modulus bilangan prima Mersenne ke-16 (M_{2203}).

III. IMPLEMENTASI

Implementasi algoritma Shamir's Secret Sharing untuk pengamanan *seed phrase* dilakukan melalui serangkaian langkah terstruktur, dimulai dari *pre-processing* daftar kata hingga eksekusi operasi aritmatika dalam *finite field*. Pendekatan sistem memprioritaskan integritas data melalui penggunaan modulus bilangan prima Mersenne berukuran besar, yang dirancang spesifik untuk menampung representasi numerik dari frasa standar BIP-39 tanpa risiko *overflow*. Seluruh logika kriptografi dibangun menggunakan bahasa pemrograman Python dan dioperasikan melalui antarmuka *Command Line Interface* (CLI).

A. Pemilihan Bilangan Prima

Sebelum proses *encoding* terhadap *seed phrase* dilakukan, diperlukan analisis untuk menentukan kebutuhan kapasitas penyimpanan yang mampu merepresentasikan seluruh kemungkinan *seed phrase* yang valid. Analisis ini bertujuan untuk memastikan bahwa skema representasi numerik dan modulus yang digunakan cukup besar untuk menampung seluruh frasa tanpa kehilangan informasi.

Varian *seed phrase* terpanjang terdiri dari 24 kata, dengan daftar kata yang telah ditetapkan oleh standar protokol BIP-39. Berdasarkan hasil analisis terhadap daftar kata tersebut, diketahui bahwa kata terpanjang memiliki panjang 8 karakter. Informasi ini digunakan sebagai dasar dalam perhitungan kebutuhan kapasitas penyimpanan dan penentuan skema representasi data yang sesuai.

```
PS C:\Users\DIERO PURNAMA\Documents\CODING\Kripto\Makalah-Kripto> python -u "c:\Users\DIERO
PURNAMA\Documents\CODING\Kripto\Makalah-Kripto\shamir\bip39\analysis.py"
The longest word in the BIP39 standard is abstract (8 characters).
3 characters: 103 words
4 characters: 442 words
5 characters: 555 words
6 characters: 508 words
7 characters: 352 words
8 characters: 88 words
```

Gambar 3.1 Analisis Daftar Kata BIP-39

Dalam skenario terburuk, sebuah *seed phrase* terdiri dari 24 kata yang masing-masing memiliki panjang 8 karakter, ditambah dengan 23 karakter spasi sebagai pemisah antar kata. Dengan demikian, estimasi panjang total *string* maksimum yang harus diproses oleh sistem adalah $(24 \times 8) + 23 = 215$ karakter.

Sistem menggunakan pengodean standar UTF-8 di mana setiap karakter direpresentasikan oleh 8 bit data. Oleh karena itu, kebutuhan memori maksimum untuk merepresentasikan satu *seed phrase* utuh ke dalam bentuk bilangan bulat adalah $215 \times 8 = 1.720$ bit.

Syarat utama dari algoritma SSS adalah nilai prima p yang dipilih harus lebih besar dari semua kemungkinan nilai rahasia S $(p > S)$ [4]. Jika modulus yang digunakan lebih kecil dari data rahasia, frasa yang direkonstruksi akan berbeda dan akan memberikan hasil yang tidak sesuai. Berdasarkan perhitungan kebutuhan 1.720 bit diatas, dibutuhkan bilangan prima dengan nilai yang lebih besar dari $2^{1720} - 1$.

Oleh karena itu, penelitian ini menggunakan bilangan prima Mersenne ke-16, yaitu $2^{2203} - 1$. Meskipun terdapat selisih ruang 483 bit yang tidak terpakai, pemilihan bilangan ini memberikan keuntungan efisiensi teoritis dalam operasi aritmatika. Karena bentuknya $2^n - 1$, operasi reduksi modular dapat disederhanakan menggunakan operasi logika bitwise yang secara komputasional lebih ringan dibandingkan operasi pembagian panjang standar.

B. Representasi Data

Salah satu prasyarat dalam penerapan skema Shamir's Secret Sharing pada *seed phrase* adalah mengonversi data yang berbentuk *string* panjang ke dalam format numerik yang sesuai untuk diproses lebih lanjut. *Seed phrase* (S) yang berupa *string* terlebih dahulu di-*encode* menggunakan standar UTF-8, kemudian hasil pengodean byte tersebut diubah menjadi representasi heksadesimal dan selanjutnya dikonversi ke bilangan bulat berbasis desimal. Proses ini memastikan bahwa seluruh karakter dalam frasa, termasuk spasi, tetap terjaga tanpa kehilangan informasi.

C. Skema Pembentukan Share

Setelah *seed phrase* dikonversi representasi integer S, proses selanjutnya adalah pembagian rahasia menggunakan skema ambang (t, n). Implementasi proses ini dilakukan dalam fungsi *make_random_shares* yang menerima parameter input berupa rahasia, ambang batas minimum (t), dan jumlah total bagian (n).

Alur algoritma pembentukan *share* adalah sebagai berikut:

1. Pembangkitan Koefisien Polinomial
   Sistem membentuk polinomial acak $f(x)$ berderajat $t - 1$. Rahasia S ditetapkan sebagai konstanta a_0. Untuk koefisien lainnya ($a_1, \dots, a_{t-1}$), sistem menggunakan *Cryptographically Secure Pseudo-Random Number Generator* (CSPRNG) yang disediakan oleh modul secrets Python. Fungsi *secrets.randbelow* digunakan untuk menghasilkan koefisien yang berada dalam rentang $[0, p - 1]$ yang aman secara kriptografis.

2. Evaluasi Fungsi
   Untuk menghasilkan n buah *share*, sistem melakukan iterasi dari $x = 1$ hingga $x = n$. Pada setiap iterasi, nilai y dihitung dengan mengevaluasi polinomial $f(x)$ menggunakan metode Horner. Metode ini dipilih karena efisien secara komputasi. Rumus yang digunakan adalah sebagai berikut:

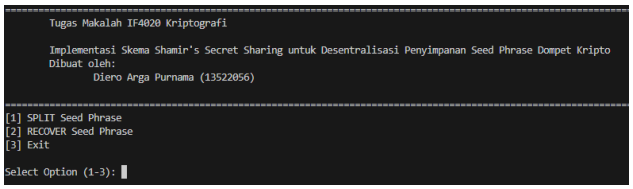
$$y_i = (\ldots((a_{t-1}x + a_{t-2})x + \cdots)x + a_0 \pmod{p} \quad (5)$$

3. Output Data

   Hasil akhir dari proses ini adalah sekumpulan pasangan titik (x , y) yang merepresentasikan sebuah *share*. Setiap pasangan ini kemudian dapat disimpan atau didistribusikan kepada pemegang hak akses.

E. Antarmuka Program

Pada saat program dijalankan, pengguna akan dihadapi dengan menu utama yang memuat opsi untuk membagi *seed phrase*, memulihkan *seed phrase*, atau keluar dari program.



Gambar 3.2 Tampilan Antarmuka Utama Program

Sebagaimana terlihat pada Gambar 3.2, pengguna akan diminta untuk memilih mode operasi melalui input numerik. Desain CLI dipilih untuk meminimalkan kompleksitas dan ketergantungan grafis, sehingga program dapat dijalankan dengan mudah pada berbagai lingkungan.

F. Skema Rekonstruksi Rahasia

Proses pemulihan *seed phrase* bertujuan untuk mengembalikan nilai integer S dari sekumpulan *share* yang dikumpulkan pengguna. Proses ini diimplementasikan dalam fungsi *recover_secret*.

Tahapan rekonstruksi dilakukan melalui langkah-langkah berikut:

1. Validasi Input

   Sistem pertama-tama memverifikasi jumlah *share* yang dimasukkan pengguna. Jika jumlah *share* kurang dari 2, proses akan dihentikan karena rekonstruksi tidak mungkin dilakukan.

2. Interpolasi Lagrange

   Sistem menghitung nilai S (atau $f(0)$) menggunakan rumus Interpolasi Lagrange. Untuk setiap titik j dalam input, sistem menghitung basis $L_j(0)$ yang melibatkan operasi perkalian dan pembagian modular.

   Untuk menghitung basis, sistem akan menggunakan persamaan (4). Pembilang a dan penyebut b akan dihitung secara terpisah. Setelah itu, algoritma euclidean akan digunakan untuk menemukan invers multiplikatif dari penyebut terhadap modulus p, sehingga operasi pembagian $\frac{a}{b}$ dapat digantikan dengan operasi perkalian $a \times b^{-1} \pmod{p}$.

   Proses ini akan diulang hingga didapatkan nilai $L_j(0)$ untuk setiap titik j. Setelah itu, nilai tersebut akan dimasukkan ke dalam persamaan (5) untuk menghitung nilai S.

3. *Decoding* Data

   Nilai integer S yang didapatkan dari hasil interpolasi kemudian dikonversi kembali menjadi format heksadesimal. Setelah itu heksadesimal tersebut akan dikonversi lagi menjadi *string* UTF-8, yang merupakan *seed phrase* asli milik pengguna. Jika terjadi kesalahan dalam input *share* (misalnya *share* palsu atau jumlah *share* yang dimasukkan kurang dari ambang batas yang ditentukan), hasil dekode akan berupa data sampah yang tidak dapat dibaca.

IV. HASIL PENGUJIAN DAN ANALISIS

Bab ini memaparkan hasil pengujian dan analisis kinerja dari implementasi perangkat lunak skema Shamir's Secret Sharing yang telah dikembangkan. Rangkaian pengujian dirancang untuk memvalidasi tiga aspek fundamental sistem, yaitu: fungsionalitas rekonstruksi kunci, keamanan mekanisme ambang batas, serta kapasitas modulus aritmatika *Finite Field* dalam menangani representasi numerik *seed phrase*. Seluruh skenario pengujian dijalankan menggunakan berbagai variasi parameter input untuk memastikan ketangguhan sistem (*robustness*), baik dalam kondisi operasional ideal maupun pada *edge cases*.

A. Pengujian Fungsionalitas Utama

Pengujian tahap pertama bertujuan untuk memverifikasi kebenaran logika algoritma dalam melakukan pembagian (*splitting*) dan pemulihan (*recovery*) *seed phrase*. Skenario ini mensimulasikan penggunaan normal di mana pengguna memiliki jumlah *shares* yang tersedia k yang memenuhi syarat ambang batas yang ditentukan ($k \geq t$).

Pada pengujian ini, digunakan parameter skema $(3, 5)$ yang berarti rahasia dibagi menjadi 5 bagian dan membutuhkan minimal 3 bagian untuk direkonstruksi. Input yang digunakan adalah *seed phrase* standar BIP-39 yang terdiri dari 12 kata, yaitu:

"*abandon ability able about above absent absorb abstract absurd abuse access accident*"

Gambar 4.1. berikut memperlihatkan proses pembagian frasa diatas menjadi 5 bagian berbeda.

Gambar 4.1 Tangkapan Layar Fase *Split* dengan Skema (3, 5)

Selanjutnya, dilakukan pengujian pemulihan untuk membuktikan bahwa data dapat dikembalikan ke bentuk semula. Dari 5 *share* yang dihasilkan sebelumnya, dipilih 3 *share* untuk dimasukkan kembali ke dalam sistem. Gambar 4.2 memperlihatkan hasil eksekusi proses rekonstruksi tersebut.

Gambar 4.2 Tangkapan Layar Fase *Recovery* Sukses

Berdasarkan hasil yang ditunjukkan pada Gambar 4.2, sistem berhasil memproses input *share* parsial dan mengeluarkan *seed phrase* yang identik dengan input awal. Keberhasilan ini memvalidasi bahwa implementasi fungsi interpolasi Lagrange pada aritmatika modular sistem berfungsi dengan benar. Sistem terbukti mampu memulihkan nilai konstanta polinomial $f(0)$ secara presisi tanpa mengalami kehilangan informasi, selama jumlah *share* yang diberikan memenuhi ambang batas minimum.

B. Pengujian Keamanan Ambang Batas

Pengujian ini bertujuan untuk memvalidasi karakteristik keamanan utama dari algoritma Shamir's Secret Sharing, yaitu ketidakmungkinan rekonstruksi rahasia jika jumlah *shares* yang tersedia k berada di bawah ambang batas ($k < t$). Skenario ini mensimulasikan situasi di mana terdapat pihak yang tidak berwenang berhasil mendapatkan sebagian kecil dari total *share* yang didistribusikan.

Menggunakan parameter skema yang sama dengan pengujian sebelumnya (3, 5), percobaan rekonstruksi dilakukan dengan sengaja hanya memasukkan 2 buah *share* ke dalam sistem.

Gambar 4.3 Tangkapan Layar Fase *Recover* Gagal

Seperti yang dapat dilihat pada Gambar 4.3, sistem gagal mengembalikan *seed phrase* semula. Hal ini terjadi karena nilai $f(0)$ yang dihasilkan oleh interpolasi Lagrange merupakan nilai acak yang tidak valid. Ketika sistem mencoba mengonversi kembali nilai tersebut menjadi teks, proses terhenti akibat kegagalan pengodean (*decoding error*). Hal ini disebabkan karena urutan *byte* yang dihasilkan dari interpolasi tersebut tidak memenuhi standar format karakter UTF-8 yang valid. Kegagalan sistem untuk menerjemahkan hasil perhitungan menjadi teks yang terbaca ini memvalidasi keamanan skema, yang membuktikan bahwa tanpa jumlah *share* yang cukup, hasil rekonstruksi hanyalah data biner acak yang tidak bermakna.

C. Pengujian Kapasitas Modulus

Pengujian terakhir ini bertujuan untuk memastikan bahwa bilangan prima yang dipilih ($2^{2203} - 1$) mampu menangani *seed phrase* dengan panjang maksimum tanpa mengalami *overflow*. Skenario ini menggunakan input *seed phrase* sepanjang 24 kata dengan total karakter yang mencapai batas maksimum teoritis (1.720 bit).

Seed phrase uji yang digunakan adalah:

"*abstract accident acoustic announce artefact attitude bachelor broccoli business category champion cinnamon congress consider convince cupboard daughter december decorate decrease describe dinosaur disagree discover*"

Gambar 4.4 Tangkapan Layar Fase *Split* Frasa Panjang

Selanjutnya, dilakukan proses pemulihan menggunakan *share* yang telah dihasilkan. Gambar 4.5 menunjukkan bahwa sistem berhasil merekonstruksi kembali frasa panjang tersebut secara utuh.

Gambar 4.5 Tangkapan Layar Fase *Recovery* Frasa Panjang

Keberhasilan rekonstruksi yang dapat dilihat pada Gambar 4.5 memvalidasi hipotesis bahwa kapasitas yang disediakan oleh bilangan prima Mersenne ke-16 cukup untuk memproses seluruh variasi kata dalam standar BIP-39. Hal ini menjamin bahwa nilai representasi integer dari rahasia S selalu lebih kecil dari bilangan prima p, sehingga tidak terjadi reduksi modular yang dapat menyebabkan kehilangan informasi asli secara permanen.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa integrasi algoritma Shamir's Secret Sharing dengan aritmatika *Finite Field* berbasis bilangan Prima Mersenne ke-16 ($2^{2203}-1$) merupakan pendekatan yang sangat efektif untuk mengamankan *seed phrase* dompet kripto. Hasil eksperimen memperlihatkan bahwa kapasitas 2.203-bit yang disediakan oleh bilangan prima tersebut memadai untuk menampung representasi numerik dari *seed phrase* terpanjang standar BIP-39 tanpa risiko *overflow*. Pemilihan parameter ini memastikan bahwa operasi aritmatika pada bilangan bulat yang sangat besar dapat dilakukan secara presisi dan efisien.

Implementasi ini memberikan kontribusi nyata dalam penyediaan mekanisme penyimpanan terdesentralisasi bagi ekosistem *Decentralized Finance* (DeFi) untuk menanggulangi risiko *Single Point of Failure*. Pergeseran dari penyimpanan pada media tunggal menuju penyimpanan terdistribusi menciptakan ketahanan data yang tidak lagi bergantung sepenuhnya pada keberadaan satu media penyimpanan, sehingga kehilangan atau kerusakan terhadap media tersebut tidak secara langsung menyebabkan hilangnya data secara permanen.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga makalah ini dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen pengampu mata kuliah IF4020 Kriptografi atas segala bimbingan dan ilmu yang telah diberikan. Ucapan terima kasih juga penulis sampaikan kepada keluarga dan teman-teman yang senantiasa memberikan dukungan selama proses penyusunan makalah ini.

PRANALA KODE PROGRAM

Kode program dapat diakses melalui repositori github berikut.

https://github.com/DieroA/Makalah-Kripto

PRANALA VIDEO DEMO

Video demo program dapat diakses melalui link berikut.

https://youtu.be/9TySLAZnoas

DAFTAR PUSTAKA

[1] Indodax, "Panduan Lengkap: Memahami dan Menjaga Keamanan Seed Phrase di Dunia Kripto". 2024. Diakses pada tanggal 14 Desember 2025. https://indodax.com/academy/panduan-keamanan-seed-phrase/

[2] Coinbase, "What is a seed phrase?". 2025. Diakses pada tanggal 14 Desember 2025. https://www.coinbase.com/learn/wallet/what-is-a-seed-phrase

[3] Blockplate, "BIP 39 Wordlist". 2025. Diakses pada tanggal 14 Desember 2025. https://www.blockplate.com/pages/bip-39-wordlist

[4] Munir, Rinaldi, "Skema Pembagian Data Rahasia (Secret Sharing Scheme)". 2025. Diakses pada tanggal 14 Desember 2025. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/37-Skema-Pembagian-Data-Rahasia-2025.pdf

[5] Wolfram Mathworld, "Mersenne Number". 2025. Diakses pada tanggal 14 Desember 2025. https://mathworld.wolfram.com/MersenneNumber.html

[6] Wolfram Mathworld, "Mersenne Prime". 2025. Diakses pada tanggal 14 Desember 2025. https://mathworld.wolfram.com/MersennePrime.html

[7] GeeksForGeeks, "Lagrange Interpolation Formula". 2025. Diakses pada tanggal 14 Desember 2025. https://www.geeksforgeeks.org/maths/lagrange-interpolation-formula/

[8] Munir, Rinaldi, "Elliptic Curve Cryptography (ECC) – Bagian 1". 2025. Diakses pada tanggal 15 Desember 2025. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/24-ECC-Bagian1-2025.pdf

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Desember 2025

Diero Arga Purnama (13522056)